Exploiting Coalition in Co-Evolutionary Learning

Yeon-Gyu Seo
Department of Computer Science
Yonsei University
134 Shinchon-dong, Sudaemoon-ku
Seoul 120-749, Korea
kitestar@candy.yonsei.ac.kr

Sung-Bae Cho
Department of Computer Science
Yonsei University
134 Shinchon-dong, Sudaemoon-ku
Seoul 120-749, Korea
sbcho@csai.yonsei.ac.kr

Xin Yao
School of Computer Science
The University of Birmingham
Edgbaston, Birmingham B15 2TT
United Kingdom
x.yao@cs.bham.ac.uk

Abstract- Adaptive behaviors often emerge through interactions between adjacent neighbors in dynamic systems, such as social and economic systems. In many cases, an individual's behaviors can be modeled by a stimulus-response system in a dynamic environment. In this paper, we use the Iterated Prisoner's Dilemma (IPD) game, which is simple yet capable of dealing with complex problems, to model a dynamic system such as social or economic systems. We investigate coalitions consisting of many players and their emergence in a co-evolutionary learning environment. We introduce the concept of confidence for players in a coalition and show how such confidences help to improve the generalization ability of the whole coalition. Experimental results will be presented to demonstrate that co-evolutionary learning with coalitions and player confidences can produce IPD game-playing strategies that generalize well.

1 Introduction

Conventional machine learning techniques in artificial intelligence have been focused on system optimization. They may not be suitable for studying adaptive behaviors in a dynamic system. An individual's behaviors in a social or economic system are dynamic, complex and often very difficult to formalize and understand. The game-theoretic approach has been used extensively in the study of social and economic systems [1]. However, only relatively simple models have been dealt with in order to make the models mathematically tractable. The evolutionary approach provides an excellent alternative to modeling and understanding more realistic and complex dynamic systems because it makes few assumptions about the model studied. It relies heavily on computational simulations rather than mathematical proofs.

The Iterated Prisoner's Dilemma (IPD) game and its variants have been used widely in modeling various social and economic phenomena [1]. It has also been used to study co-evolutionary learning under various conditions [3, 4], where the primary purpose is not to model a dynamic system but to study how co-evolution can be used in learning novel strategies [5, 8, 9]. For example, co-evolutionary learning of strategies for $N(N > 2)$ player IPD games has been studied extensively [3, 4]. The impact of the group size N and the history length of the N player game on the evolved strategies was also studied. Recently, a new approach to automatic design of modular learning systems has been proposed based on a combination of co-evolutionary learning and fitness sharing [8, 9]. The IPD games with multiple choices of cooperation levels and reputation for each player have introduced a new dimension to the study of co-evolutionary learning and brought the IPD game one step closer to the real life situations [10].

In our previous work on evolving coalitions in the IPD game, we focused on the issue of local interaction [11, 12]. This paper investigates the issue of generalization in a co-evolutionary learning system using the IPD game as an example. This issue was first closely examined in [5] using the classical 2 player IPD game. We will show in this paper how the generalization ability of a coalition can be improved through player's confidences.

A coalition, motivated to get more payoff or survive for long time, consists of better performing players extracted from the population. Each player is assigned a confidence within the coalition, which influences how much say the player has in determining the next move for the coalition. Since different players may be better at dealing with different opponents [9], they should have different confidences in dealing with different opponents. In this paper, confidences of players are adapted through evolutionary learning automatically. They are not pre-defined or fixed.

The rest of this paper is organized as follows: section 2 introduces the IPD game and the co-evolutionary approach to it. The representation of IPD game strategies is given. Section 3 explains strategy coalitions in the IPD game and how they are formed in evolution. Section 4 introduces the concept of player confidence and describes how to evolve player confidences within a coalition so that the generalization ability of the whole coalition can be improved. Section 5 presents the experimental results. Finally, section 6 concludes the paper with a brief summary of the paper and a few remarks.

2 Evolutionary Approach to Iterated Prisoner's Dilemma Games

2.1 Iterated Prisoner's Dilemma Games

In the 2 player IPD game [1], each player can choose one of the two choices, defection (D) or cooperation (C). Table 1

0-7803-6375-2/00/$10.00 ©2000 IEEE.

shows the payoff matrix of the 2 player IPD game. The game is non-zerosum and non-cooperative. One player's gain may not be the same as the other player's loss. There is no communications between the two players. The game can be repeated infinitely. None of the players knows when the game is supposed to end.

Table 1: Payoff matrix of the 2IPD game. T > R > P > S, 2R > T + P

	Cooperate	Defect
Cooperate	R	S
Defect	T	P

According to the conditions given in Table 1, if the game is played for one round only, the optimal strategy is definitely defection (D). However, if the game is played for many rounds, mutual defection (D) may not be the optimal strategy. Mutual cooperation (C) will perform better than mutual defection for iterated IPD games. There has been a great deal of work in game theories investigating the optimal strategy for the 2 player game under various conditions [1]. The focus of this paper will be on co-evolutionary learning and the generalization ability of learned strategies.

2.2 Representation of Strategies

The evolutionary approach to strategy learning for the 2IPD game was popularized by Axelrod [2]. One of the most important issues in evolving game-playing strategies is their representation. There are two different possible representations for N player IPD games ($N > 2$) [3, 4]. One is based on a generalization of Axelrod's representation scheme for the 2 player game. For the N player game, this scheme encodes additional information that should not be made available to players. It also scales poorly. Another representation scheme avoids these two problems and has been used widely for studying N player games [3, 4]. For 2 player games, these two representations are equivalent. This paper only uses the 2 player game.

Each genotype in our representation is a lookup table that covers every possible history of the last few steps. History in a game is represented as a binary string of $2l$ bits, where the first l bits represent the player's own previous l actions (most recent to the left, oldest to the right), and the other l bits represent the previous actions of the other player. For example, during a game with a remembered history of 2 steps, i.e., $l = 2$, one player might see this history:

$l = 2$: Example history 11 01

The first l bits, 11, means this player has defected (an '1') for both of the previous $l = 2$ steps, while the opponent has cooperated (0) on the most recent step, and defected (1) on the step before, as represented by 01.

For the game remembering l previous steps, there are 2^{2l} possible histories. The genotype therefore contains an action (cooperate "0", or defect "1") for each for these possible histories. So we need at least 2^{2l} bits to represent a strategy. At the beginning of the game, there are no previous l steps of play from which to look up next action, so each genotype should also contain its own extra bits that define the presumed pre-game moves. The total genotype length is therefore $2^{2l} + 2l$ bits.

3 Evolving Coalition of Strategies

Each player in the IPD game can be defined by the model given in Table 2. It describes all the information available to the player in deciding his/her next move. In the Table 2, ID is assigned to each agent and unique identifier, and History is a memory to remember previous moves. Strategy is a lookup table for his next move. Each agent has these three properties but other properties are true for the agent that belongs to a coalition. BelongTo is a information of coalition that he belongs to, and Confidence is related to his status within coalition, which is determined by his rank. If confidence is high, his decision can have much effect on coalition's decision.

Table 2: Player model for the 2 player game with coalitions.

Property	Role
ID	unique identifier
History	previous moves
Strategy	next move
BelongTo	info. of coalition
Confidence	rate of participating in moves within coalition
Rank	rank within coalition

3.1 Forming Coalitions

Co-evolutionary learning may not provide robust strategies which generalize well [5]. Evolved strategies may overfit to their current population and perform poorly against good strategies that are not present in the current population [5, 9]. One effective approach to tackle this problem and improve generalization of co-evolutionary learning is to automatically learn a group of cooperative strategies, each of which is a specialist in dealing with a certain type of opponents, so that the whole group becomes very robust generalizes well [8, 9]. Since such groups and specialists within a group are evolved, not partitioned manually, this approach can be regarded as a method of automatic design of modular systems, where each module consists of similar specialists [8, 9].

In this paper, a different method for evolving a group of strategies (i.e., a coalition in this paper) is proposed. It uses the idea of confidence of an player in a coalition. A player's confidence is not pre-defined and fixed, but evolved. A player may have different confidences in dealing with different opponents. A player is allowed to be able to take part in the next move of coalition at the rate of his confidence. In other words, coalition's next move can be determined by all players' confidences.

Three conditions must be satisfied for two strategies to be in the same coalition:

1. Both strategies are better strategies (his fitness is higher than 1.2 times the average fitness of population) and the result of a game between them is harmful (his score is lower than 0.8 times the average fitness of population) to them.

2. Both strategies are better strategies and combining of them results in good interest (his score is higher than 1.2 times the average fitness of population).

3. The two strategies are different.

The IPD game with coalitions is played according to the procedure given in Figure 1. During each generation of evolution, the 2 player game is played between two randomly selected players. Round robin games are played in each generation. A player can play against a coalition. Also, there are games between coalitions. After a game, new coalition may be formed. Two players can form a coalition. A player can also join an existing coalition if its fitness is higher than the coalition's threshold. In the case of games between coalitions, there is no change in each coalition except for fitness.

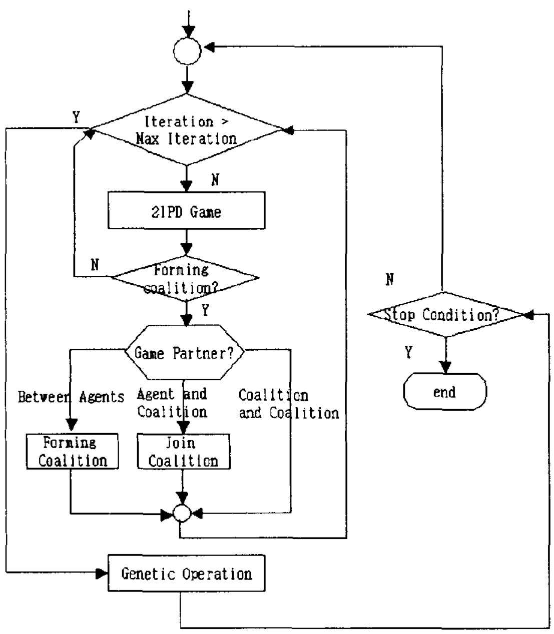

Figure 1: The 2 player IPD game with coalitions.

Once a player has joined a coalition, players within the coalition (including the new one) will play IPD games in round-robin. For k players in a coalition, a total of $k(k-1)/2$ games will be played. If the total number of players (i.e., k) is greater than a pre-defined maximum coalition size, the weakest player (in terms of the total payoff he/she obtained in all round-robin games) will be removed from the coalition, as shown in Figure 2. All players within a coalition are ranked (sorted) according to the total payoff he/she obtained in all round-robin games. A player's confidence is assigned in proportion to his/her ranking in the coalition. Here, sum of confidences is 1. For example, if there are four players, confidences will be 0.8, 0.9 1.1 1.2. The confidence plays an important role in determining the player's impact on the coalition's next move.

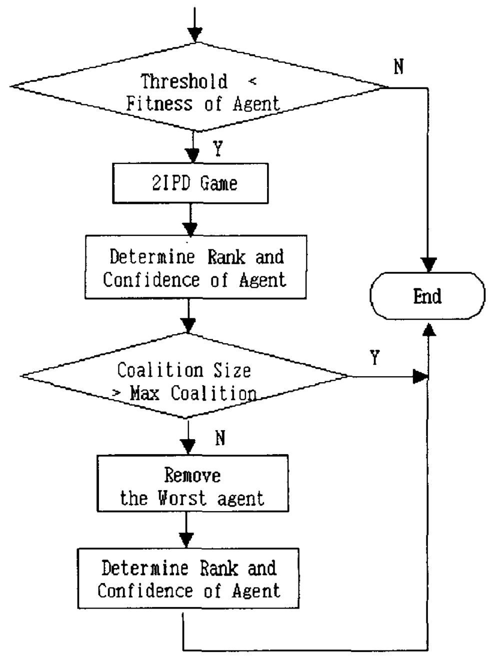

Figure 2: Coalition forming during evolution.

Figure 3 shows how to combine confidences of different players in a coalition to produce the coalition's next move. In Figure 3, O indicates opponent's moves and *His* indicates coalition's moves in previous steps. In the figure, A,B,C and D are agents within a coalition and A_c and A_d is agent's next move, reflected his confidence, for given history. For example, if A's next move is cooperation, A_c is $1 \times$ A's confidence and A_d is 0. There is no weight in input to agent.

3.2 Evolution of Coalitions

Coalitions may be formed or removed from a population during evolution. In our experiments, coalition below the average fitness of players in a population will be removed. In addition to coalition formation as described previously, new coalition may also be generated by crossover of existing coalitions. Crossover of coalitions is done by exchanging same number of players, as shown in Figure 4.

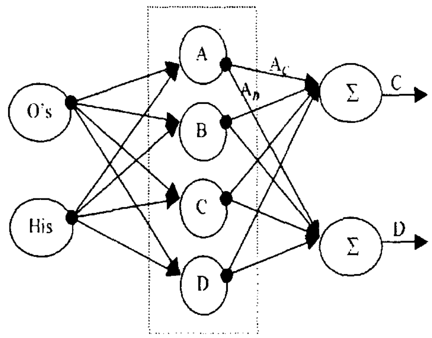

Figure 3: Combining confidences of multiple players within coalition.

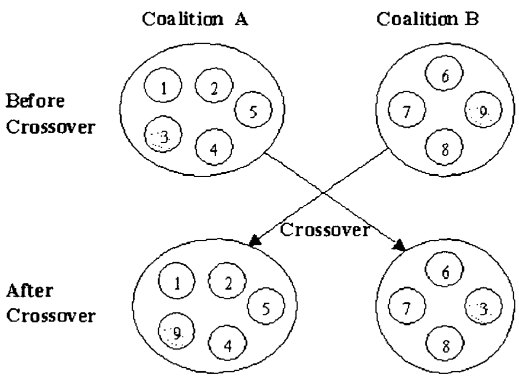

Figure 4: Crossover used in evolving coalitions.

It is important to point out that a player will only exist in the coalition once it has joined in. As a result, only weak players will be in the population without belonging to any coalitions (separate from the population). Strong players will have joined coalitions. New players in the population are generated by recombining players from coalitions. We do not use weak players in the population to generate new players. Figure 5 shows how to generate new players from those already in coalitions.

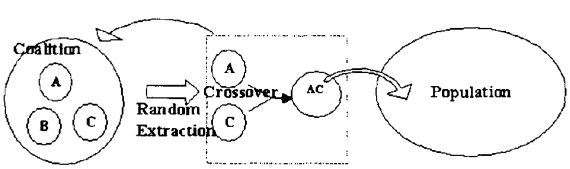

Figure 5: Generating new players by mixing players in coalitions.

4 Gating Strategies in Coalitions

Each player has a confidence to take part in deciding coalition's next move. If the confidence is fixed, a coalition will have fixed behaviors regardless what kind of opponents it may have. Such a coalition can be represented by a single player and thus is not very interesting for us. We would like to adjust player's confidence dynamically for different opponents so that the coalition can make best use of different players within the coalition according to different types of opponents. This motivation is the same as that discussed in some previous work [13].

4.1 Adjusting Player Confidences

To improve the generalization ability of a coalition, techniques such as opponent modeling and gating can be used. Opponent modeling can be used to model and predict opponent's next move. Such prediction, if sufficiently accurate, can help the coalition to choose the optimal move. However, it is difficult to model opponent's strategy accurately. Darwen and Yao [8, 9] adopted a modularization approach to pool different specialist players together and then used a gating algorithm to determine which specialist player should respond to an unseen opponent.

In this paper, we use player confidences to decide which move should be taken by the coalition against an opponent. Population consists of coalitions that have different confidences, and coalition has same players as other players that are obtained during the evolution. A coalition has a history table for its own and its opponent's moves and can use this information to change player confidences dynamically. Figure 6 shows how player confidences are adjusted during evolution. Player confidences in a coalition are first randomly initialized as real numbers between 0 and 2. A condfidence table contains player confidences for all possible combinations of history. The training set for adjusting player confidences consists of several well known game-player strategies, such as TFT, Trigger, CDCD, etc [6, 7].

During evolution, confidences performing well are selected from coalitions. Crossover mixes confidences between coalitions in the population. Mutation changes a specified confidence into a random real number between zero and two.

5 Experimental Results

The first experiment is designed to evolve coalitions using co-evolutionary learning and the second is to evolve player confidences so that coalitions can play adaptively against different opponents. In our experiments, the payoff matrix used follows the one given in [5].

5.1 Evolving Coalitions

In our experiments, we used population size 50, crossover rate 0.6, mutation rate 0.001 and rank-based selection is used.

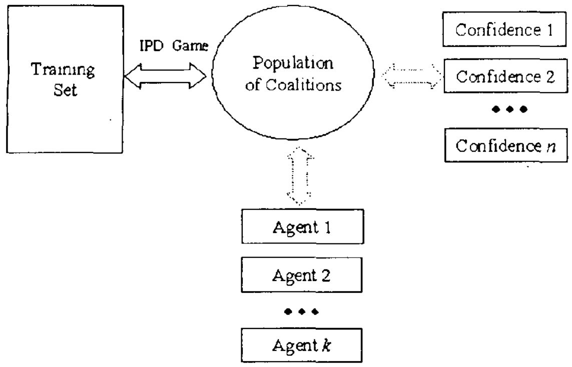

Figure 6: The evolution of confidences.

A genetic algorithm with one-point crossover and elitism was adopted [14]. For the 2 player game, the history length was 2. The maximum coalition size was10 and the maximum number of coalitions is one third of the population size (i.e.,16). This is because there may remain few players in the populations, when coalition For the experiments presented in this subsection, player confidences are fixed and not evolved. The evolution of player confidences will be considered in the next subsection.

Figure 7 shows the average fitness of coalitions for 4 runs during evolution. In the beginning of evolution, the average fitness of coalitions were well above the average fitness of players in the population. However, the difference decreased as time went by, because players in the population had gradually learned to deal with stronger coalitions. (Bear in mind that new players in the population were generated by recombining strong players from coalitions.). In other words, players in the population also gradually evolve to adapt to their environment.

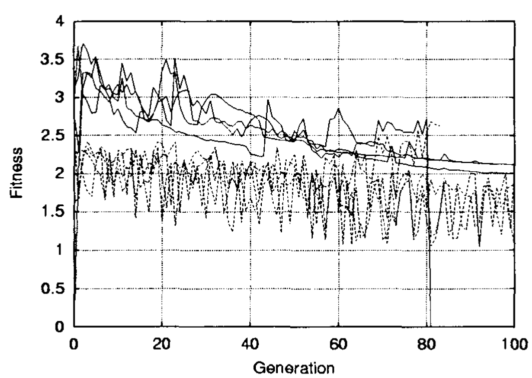

Figure 7: Average fitness of coalitions and players in the population. Solid lines are for coalitions and dashed lines for players.

Figure 8 shows similarity (i.e., convergence) among players in the population. As shown by the figure, the population had more or less converegedafter 100 generations. Coalitions may start disappearing from the population when their fitness is below individual players' fitness.

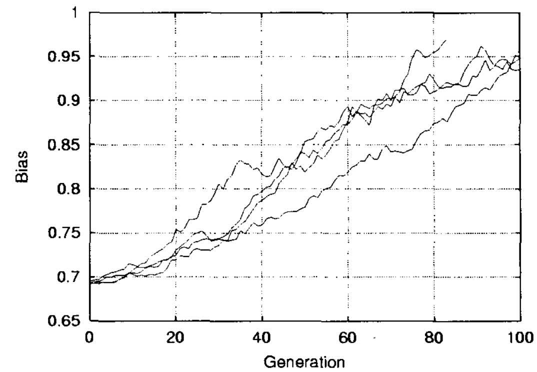

Figure 8: Population convergence.

In the above experiments, the coalition's next move is determined by players' weighted voting at a fixed rate (weight,here, corresponds to confidence). This means that a coalition's strategy against an opponent can be represented by a single player. A coalition may not be as fit as it should have been in comparison with individual players not in the coalition. Figure 9 shows the number of coalitions during evolution. The number of coalitions could become very small or even zero as a result of fixed player confidences.

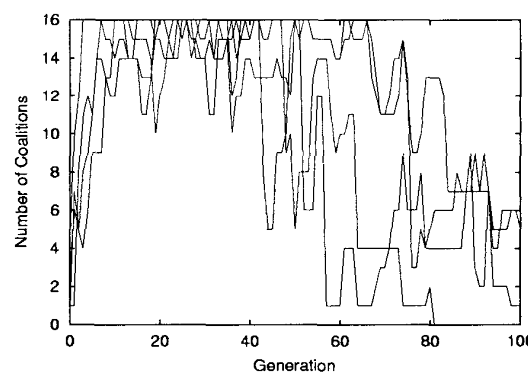

Figure 9: The number of coalitions in the evolution.

5.2 Evolving Confidences

In this experiment, we used the same experimental setup as before except that (μ,λ) selection was used. The training set used in evolving player confidences consists of seven well-known strategies and a random strategy, as shown by Table 3.

Figures 10 to 15 show the average fitness of evolved coalitions and test players when they played against each other. It is clear that the coalitions learned to cooperate with cooperators (including TFT, Trigger and TF2T), retaliate against

Table 3: Training set for evolving player confidences.

Strategy	Characteristics
TFT	initially cooperate, and then follow opponent
Trigger	initially cooperate once opponent defects, continuously defect
TF2T	similar TFT, but defect for opponent's 2 defection
AllD	always defect
CDCD	cooperate and defect over and over
CCD	cooperate and cooperate and defect
C10DAll	cooperate before 10 rounds and always defect after that
Random	random move

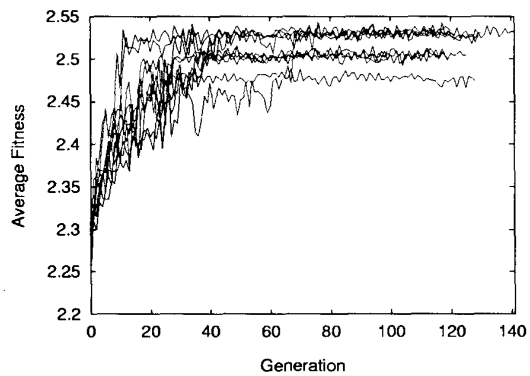

Figure 10: Average fitness of coalitions.

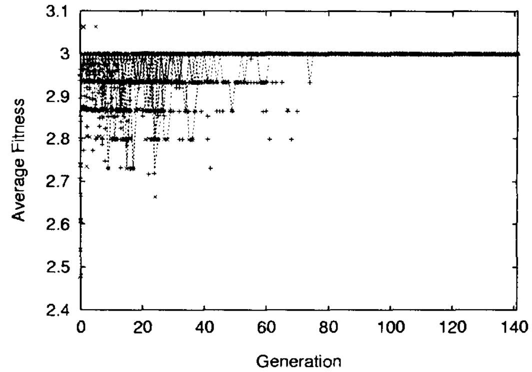

Figure 11: Average fitness of cooperators.

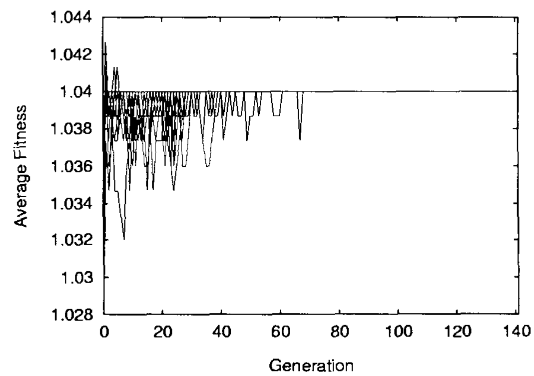

Figure 12: Average fitness of AllD Strategy.

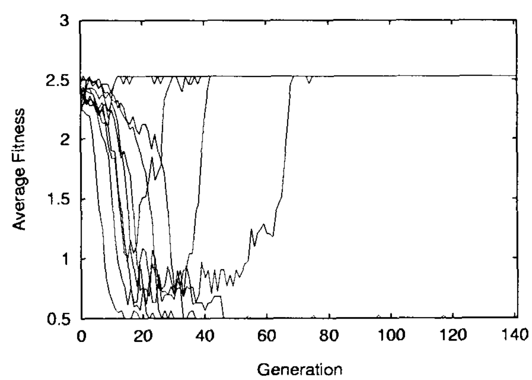

Figure 13: Average fitness of CDCD strategy.

AllD, and beat the random player soundly. The coalitions also played at least as well as C10Dall and CDCD by defeating or tying with them. Regardless of opponents, the coalitions could consistently achieve a high fitness value.

To evaluate the generalization ability of coalitions with evolving confidences, we have selected 30 top ranked players in a random population of 300 as the test set. Each player in the test set then plays against evolved coalitions in round-robin games for ten times. The test set is given by Table 4.

In the test of generalization ability of coalitions, we allowed player confidences to vary with different opponents. Table 5 shows the performance of a player against the 30 test players. The second and third columns in Table 5 indicate that there is a significant improvement in coalitions' performance when player confidences are allowed to evolve. The second column indicates the results without the evolution of confidences and the third column indicates those with it. In general, coalitions' winning percentage has increased significantly and the losing percentage decreased. In comparison

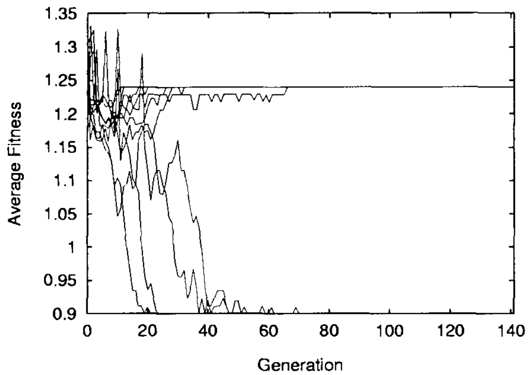

Figure 14: Average fitness of C10Dall strategy.

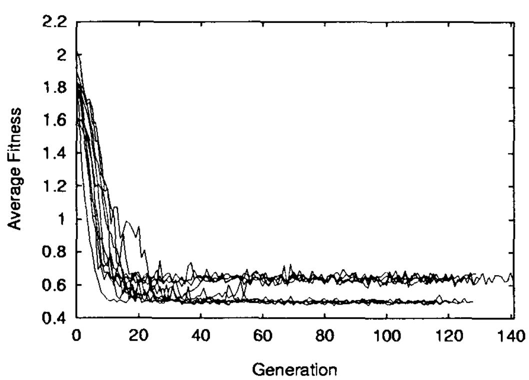

Figure 15: Average fitness of the random strategy.

Table 4: 30 opponents in the test set, which were best performers from a population of 300 random players.

History	Lookup Table	History	Lookup Table
1000	0111101111101110	1111	1101111001110011
0100	1101111111110011	1100	0101100111010001
1000	1111111000000111	0111	0011111101010010
0111	0111111110110111	1111	0111011101010111
1100	0111001101010011	1111	0101101100111100
0001	1011111001011100	0111	1001110101010110
0001	0011010111111110	1101	0001000110011010
1100	0001110110110010	0010	1101110011111101
1000	1101110011011101	1001	1001100101011000
0010	1101101111000110	1000	1101000111101010
1111	1111101100011011	0110	1001011001110110
0110	0111010011011111	1011	0101111101110010
1010	1011010111111100	0001	0011110111011000
1101	1101000101011110	1110	0111010101110001
1100	1101011110111011	1011	0101011111110100

with players in the training set, the coalitions also performed quite well against the 30 test players. They only performed worse than AllD on average. They seemed to be more likely to tie a game than those players in the training set. This appears to indicate that the coalitions have learned not to lose a game as its top priority.

Table 5: Performance against opponent strategies

Strategy	Wins	Ties	Avg.	Opp. Avg.
Before	8.64±4.9	6±2.19	1.84±0.28	1.75±0.59
After	18.55±0.5	4±0.63	2.16±0.07	0.92±0.29
TFT	8	0	1.70	1.77
Trigger	30	0	2.13	0.80
TF2T	7	0	1.54	2.40
AllD	30	0	2.17	0.7
CDCD	0	0	1.05	2.75
CCD	0	0	0.91	3.34
C10Dall	27	0	1.97	1.12

We used a small fixed training set to evolve player confidences in this paper. This could lead to confidences that were over-specialized to the fixed training set. It is necessary to investigate how representative the training set is and whether it has covered all the major types of different players. This will be one of our future work. We expect incremental evolution to play a role in co-evolutionary learning in further enhancing the generalization ability of evolved coalitions.

6 Conclusions

Combining multiple players in a group can be a very effective way of designing a learning system that generalizes well [8, 9, 16], this paper further supports that this is a good approach. We have evolved coalitions consisting of a number of players in the IPD game. Each player in a coalition has a confidence which specifies how well he/she is in dealing with an opponent. The confidence is not fixed. It is different for different opponents.

Experiments have been carried out to compare players with and without adaptive confidences. It is found that using adaptive confidences can improve coalitions' performance in testing significantly. Coalitions with adaptive player confidences can deal with very different opponents, regardless whether they are cooperators or defectors.

Although we have used the 2-player IPD game in this paper, some of the results we have obtained may be applicable to more complex games. It will be very interesting to study the evolution of coalitions and confidences in a real world environment. For example, it is interesting to investigate how coalitions could be formed among different countries in the world, how coalitions could be formed among different parties in a country, how coalitions could be formed in the commercial market, etc. Also, the results obtained from simulated simple game can be applied to play real on-line game in the network such as Tron [15].

References

[1] A. M. Colman, *Game Theory and Experimental Games.* Oxford, England: Pergamon Press, 1982.

[2] R. Axelrod, "The evolution of strategies in the iterated prisoner's dilemma," in *Genetic Algorithms and Simulated Annealing* (L. Davis, ed.), ch. 3, pp. 32–41, San Mateo, CA: Morgan Kaufmann, 1987.

[3] X. Yao and P. Darwen, "An experimental study of N-person iterated prisoner's dilemma games," *Informatica*, vol. 18, pp. 435–450, 1994.

[4] X. Yao and P. J. Darwen, "An experimental study of N-person iterated prisoner's dilemma games," in *Progress in Evolutionary Computation, Lecture Notes in Artificial Intelligence, Vol. 956* (X. Yao, ed.), (Heidelberg, Germany), pp. 90–108, Springer-Verlag, 1995.

[5] P. J. Darwen and X. Yao, "On evolving robust strategies for iterated prisoner's dilemma," in *Progress in Evolutionary Computation, Lecture Notes in Artificial Intelligence, Vol. 956* (X. Yao, ed.), (Heidelberg, Germany), pp. 276–292, Springer-Verlag, 1995.

[6] R. Axelrod and D. Dion, "The further evolution of cooperation," *Science*, vol. 242, pp. 1385–1390, 9 December 1988.

[7] R. Axelrod, *The evolution of cooperation.* New York: Basic Books, 1984.

[8] P. Darwen and X. Yao, "Automatic modularization by speciation," *Proc. of 1996 IEEE International Conference on Evolutionary Computation (ICEC'96)*, May 1996, Nagoya, Japan, IEEE Press, pp.88–93.

[9] P. J. Darwen and X. Yao, "Speciation as automatic categorical modularization," *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 2, pp. 101–108, 1997.

[10] X. Yao and P. Darwen, "How important is your reputation in a multi-agent environment," in *Proceedings of the 1999 IEEE International Conference on Systems, Man, and Cybernetics (SMC'99)*, (Piscataway, NJ, USA), Vol. 2, pp. II-575–II-580, IEEE Press, October 1999.

[11] Y. G. Seo, S. B. Cho and X. Yao, "Emergence of cooperative coalition in NIPD game with localization of interaction and learning," *Proc of Congress on Evolutionary Computation (CEC'99)*, Piscataway, NJ USA), pp. 877–884, IEEE Press, 1999.

[12] Y. G. Seo, S. B. Cho and X. Yao, "The impact of payoff function and local interaction on the N-player Iterated Prisoner's Dilemma game," *Knowledge and Information Systems: An International Journal, Submitted*, 1999.

[13] X. Yao, Y. Liu and P. Darwen, "How to make best use of evolutionary learning," *Complexity International: An Electronic Journal of Complex Systems Research (ISSN 1320-0682)*, Vol. 3, July 1996. (URL: `http://www.csu.edu.au/ci/vol3/yao.` Also appeared in paper form in *Complex Systems — From Local Interactions to Global Phenomena*, IOS Press, Amsterdam, pp.229–242, 1996.

[14] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, Reading, MA, 1989.

[15] Pablo Funes, et. al, "Animal-animat coevolution : Using the animal population as fitness function," *Proceedings of International Conference on Simulation of Adaptive Behavior (SAB'98)*, MIT Press, pp. 525–533, 1998.

[16] X. Yao and Y. Liu, "Making use of population information in evolutionary artificial neural networks," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, **28**(3):417-425, June 1998.